A group of algorithms is presented generalizing the Fast Fourier Transform to the case of noninteger frequencies and nonequispaced nodes on the interval $[-\pi, \pi]$. The schemes of this paper are based on a combination of the classical Fast Fourier Transform with a version of the Fast Multipole Method, and generalize both the forward and backward FFTs. Each of the algorithms requires $O(N \cdot \log N + N \cdot \log(1/\varepsilon))$ arithmetic operations, where ε is the precision of computations and N is the number of nodes. The efficiency of the approach is illustrated by several numerical examples.

Fast Fourier Transforms for Nonequispaced Data II

A. Dutt and V. Rokhlin

Research Report YALEU/DCS/RR-980
August 1993

The authors were supported in part by the *Office of Naval Research* under Grant N00014-89-J-1527 and in part by the National Science Foundation under Grant DMS9012751.
Approved foı public release: distribution is unlimited.
Keywords: *FFT, Trigonometric Series, Fourier Analysis, Interpolation, Fast Multipole Method, Approximation Theory*

1 Introduction

Fourier techniques have been a popular analytical tool in physics and engineering for more than two centuries. A reason for this popularity is that the trigonometric functions $e^{i\omega x}$ are eigenfunctions of the differentiation operator and thus form a natural basis for representing solutions of many classes of differential equations.

More recently, the arrival of digital computers and the development of the fast Fourier transform (FFT) algorithm in the 1960s (see [7]) have established Fourier analysis as a powerful and practical numerical tool. The FFT, which computes discrete Fourier transforms (DFTs), is now a central component in many scientific and engineering applications, most notably in the areas of spectral analysis and signal processing. Numerous applications, however, involve unevenly spaced data, whereas the FFT requires input data to be tabulated on a uniform grid. In this paper, we present a collection of algorithms which overcome this limitation of the FFT while preserving its computational efficiency. These algorithms are designed for the efficient computation of certain generalizations of the DFT, namely the application and inversion of the transformation $F : \mathbf{C}^N \rightarrow \mathbf{C}^N$ defined by the formulae

$$F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j}, \tag{1}$$

for $j = 1, \ldots, N$, where $x = \{x_1, \ldots, x_N\}$ is a sequence of real numbers in $[-\pi, \pi]$ and $\alpha = \{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ is a sequence of complex numbers. The number of arithmetic operations required by each of the algorithms of this paper is proportional to

$$N \cdot \log N + N \cdot \log\left(\frac{1}{\varepsilon}\right) \tag{2}$$

where ε is the desired accuracy, compared with $O(N^2)$ operations required for the direct application and $O(N^3)$ for the direct inversion of the transformation described by (1).

Remark 1.1 The DFT in "unaliased" form is described by the formula

$$f_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{2\pi ikj/N} \tag{3}$$

for $j = -N/2, \ldots, N/2-1$, which is clearly a special case of (1). The FFT algorithm employs a sequence of algebraic manipulations to reduce the number of operations for the DFT from $O(N^2)$ to $O(N \cdot \log N)$. In the more general case of (1), the structure of the linear transformation can also be exploited via a combination of certain analytical results and the FFT.

The algorithms of this paper utilize the fact that a Fourier series is a trigonometric polynomial. When dealing with the values of this polynomial at equispaced nodes on the unit circle, the FFT can be applied. In this paper, however, we are interested in the values at nonuniformly spaced nodes, which are the values of the polynomial which interpolates the equispaced values. The algorithms we will describe rely for their efficiency on a combination of the FFT with a fast

algorithm for evaluating trigonometric polynomial interpolants which uses a version of the Fast Multipole Method (FMM) specifically designed for the geometry of the circle. This interpolation algorithm is closely related to the interpolation algorithm described in [9] for polynomials tabulated on the line.

Remark 1.2 Throughout this paper we will be using the well known Lagrange representation of polynomial interpolants. For a function $f : \mathbf{C} \to \mathbf{C}$ tabulated at nodes $z_1, \ldots, z_N$, this is defined by the formula

$$P_N(z) = \sum_{j=1}^{N} f(z_j) \cdot \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{z - z_k}{z_j - z_k}. \tag{4}$$

Following is a plan of this paper. Section 2 contains a number of results from analysis and approximation theory, and in Section 3 we describe both formally and informally how these results are used, together with the FMM, in the construction of the fast algorithms of this paper. Results of several of our numerical experiments are presented in Section 4 to demonstrate the performance of these algorithms, and finally in Section 5 we discuss several generalizations and conclusions.

Remark 1.3 An alternative approach to the problems of this paper is presented in [10], where an interpolation scheme based on the Fourier analysis of the Gaussian bell is used in place of the FMM-based interpolation scheme of the present paper. We compare the two approaches in Section 5.

2 Mathematical and Numerical Preliminaries

This section is divided into two parts. In Subsection 2.1 we present several identities which are employed in the development of the fast algorithms of this paper. Subsection 2.2 contains a collection of error bounds which allow us to perform calculations to any prescribed accuracy.

2.1 Analytical Tools

The main results of this subsection are Theorems 2.3 and 2.4 which describe linear transformations connecting the values of a Fourier series at two distinct sets of points. Lemmas 2.1 and 2.2 provide intermediate results which are used in the proofs of these theorems.

Lemma 2.1 *Let $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_N\}$ be sequences of real numbers on the interval $[-\pi, \pi]$, and let $\{w_1, \ldots, w_N\}$ and $\{z_1, \ldots, z_N\}$ be sequences of complex numbers defined by the formulae*

$$w_j = e^{ix_j} \tag{5}$$

$$z_j = e^{iy_j} \tag{6}$$

for $j = 1, \ldots, N$. Then,

$$\prod_{\substack{k=1\\k\neq j}}^{N}(w_l - z_k) = w_l^{(N-1)/2} \cdot \prod_{\substack{k=1\\k\neq j}}^{N} z_k^{1/2} \cdot 2i \cdot \sin((x_l - y_k)/2) \tag{7}$$

for $l = 1, \ldots, N$, and

$$\prod_{\substack{k=1\\k\neq j}}^{N}(z_j - z_k) = z_j^{(N-1)/2} \cdot \prod_{\substack{k=1\\k\neq j}}^{N} z_k^{1/2} \cdot 2i \cdot \sin((y_j - y_k)/2) \tag{8}$$

for $j = 1, \ldots, N$.

Proof. A sequence of simple algebraic manipulations and trigonometric identities yields

$$\begin{aligned}
\prod_{\substack{k=1\\k\neq j}}^{N}(w_l - z_k) &= \prod_{\substack{k=1\\k\neq j}}^{N}(e^{ix_l} - e^{iy_k}) \\
&= \prod_{\substack{k=1\\k\neq j}}^{N} e^{i(x_l+y_k)/2} \cdot (e^{i(x_l-y_k)/2} - e^{-i(x_l-y_k)/2}) \\
&= e^{i(N-1)x_l/2} \cdot \prod_{\substack{k=1\\k\neq j}}^{N} e^{iy_k/2} \cdot 2i \cdot \sin((x_l - y_k)/2) \qquad (9) \\
&= w_l^{(N-1)/2} \cdot \prod_{\substack{k=1\\k\neq j}}^{N} z_k^{1/2} \cdot 2i \cdot \sin((x_l - y_k)/2).
\end{aligned}$$

Substituting z_j for w_l and y_j for x_l in (9), we also obtain

$$\prod_{\substack{k=1\\k\neq j}}^{N}(z_j - z_k) = z_j^{(N-1)/2} \cdot \prod_{\substack{k=1\\k\neq j}}^{N} z_k^{1/2} \cdot 2i \cdot \sin((y_j - y_k)/2). \tag{10}$$

□

The following lemma describes an alternative representation of the well known Lagrange interpolation formula for polynomials in the case when the interpolation points lie on the circle.

Lemma 2.2 *Let $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_N\}$ be sequences of real numbers in the interval $[-\pi, \pi]$, and let $\{f_1, \ldots, f_N\}$ be a sequence of complex numbers. Further, let $\{w_1, \ldots, w_N\}$ and $\{z_1, \ldots, z_N\}$ be sequences of complex numbers defined by the formulae*

$$w_j = e^{ix_j} \tag{11}$$

$$z_j = e^{iy_j} \tag{12}$$

for $j = 1, \ldots, N$. *Then,*

$$\sum_{j=1}^{N} f_j \cdot \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{w_l - z_k}{z_j - z_k} = w_l^{N/2} \cdot c_l \cdot \sum_{j=1}^{N} f_j \cdot z_j^{-N/2} \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right) \tag{13}$$

where $\{c_l\}$ *and* $\{d_j\}$ *are defined by the formulae*

$$c_l = \prod_{k=1}^{N} \sin((x_l - y_k)/2), \tag{14}$$

$$d_j = \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{1}{\sin((y_j - y_k)/2)} \tag{15}$$

for $j, l = 1, \ldots, N$.

Proof. Dividing (7) by (8) we obtain

$$\begin{aligned} \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{w_l - z_k}{z_j - z_k} &= \frac{w_l^{(N-1)/2}}{z_j^{(N-1)/2}} \cdot \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{z_k^{1/2} \cdot 2i \cdot \sin((x_l - y_k)/2)}{z_k^{1/2} \cdot 2i \cdot \sin((y_j - y_k)/2)} \\ &= \frac{e^{-i(x_l - y_j)/2}}{\sin((x_l - y_j)/2)} \cdot \frac{w_l^{N/2} \prod_{k=1}^{N} \sin((x_l - y_k)/2)}{z_j^{N/2} \prod_{k \neq j} \sin((y_j - y_k)/2)}, \end{aligned} \tag{16}$$

and the combination of (16) with the fact that

$$\begin{aligned} \frac{e^{-i(x_l - y_j)/2}}{\sin((x_l - y_j)/2)} &= \frac{\cos((x_l - y_j)/2) + i \sin((x_l - y_j)/2)}{\sin((x_l - y_j)/2)} \\ &= \frac{1}{\tan((x_l - y_j)/2)} - i, \end{aligned} \tag{17}$$

gives us

$$\sum_{j=1}^{N} f_j \cdot \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{w_l - z_k}{z_j - z_k} = w_l^{N/2} \cdot c_l \cdot \sum_{j=1}^{N} f_j \cdot z_j^{-N/2} \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right), \tag{18}$$

where $\{c_l\}$ and $\{d_j\}$ are defined by (14) and (15). □

The following theorem provides a formula for determining the values of a Fourier series at a set of points in terms of the values of this series at another set of points.

Theorem 2.3 *Let* $\{x_1, \ldots, x_N\}$ *and* $\{y_1, \ldots, y_N\}$ *be sequences of real numbers in the interval* $[-\pi, \pi]$, *and let* $\{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ *be a sequence of complex numbers. Further, let* $\{w_1, \ldots, w_N\}$,

$\{z_1, \ldots, z_N\}$, $\{f_1, \ldots, f_N\}$ and $\{g_1, \ldots, g_N\}$ be sequences of complex numbers defined by the formulae

$$\begin{aligned} w_j &= e^{ix_j} && (19) \\ z_j &= e^{iy_j} && (20) \\ f_j &= \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{iky_j} && (21) \\ g_j &= \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j} && (22) \end{aligned}$$

for $j = 1, \ldots, N$. Then,

$$g_l = c_l \cdot \sum_{j=1}^{N} f_j \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right), \tag{23}$$

where $\{c_l\}$ are defined by (14) and $\{d_j\}$ are defined by (15).

Proof. Let the polynomial P_α be defined by the formula

$$P_\alpha(z) = \sum_{k=0}^{N-1} \alpha_{k-N/2} \cdot z^k. \tag{24}$$

The Lagrange interpolation formula relates the values of P_α at the points $\{w_l\}$ to the values at the points $\{z_k\}$ via the expressions

$$P_\alpha(w_l) = \sum_{j=1}^{N} P_\alpha(z_j) \cdot \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{w_l - z_k}{z_j - z_k} \tag{25}$$

for $l = 1, \ldots, N$, and applying Lemma 2.2 to (25) we obtain

$$P_\alpha(w_l) = w_l^{N/2} \cdot c_l \cdot \sum_{j=1}^{N} P_\alpha(z_j) \cdot z_j^{-N/2} \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right). \tag{26}$$

From the combination of (24) and (19)–(22) we see that

$$P_\alpha(z_j) = z_j^{N/2} \cdot \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot z_j^k = z_j^{N/2} \cdot f_j \tag{27}$$

and

$$P_\alpha(w_l) = w_l^{N/2} \cdot \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot w_l^k = w_l^{N/2} \cdot g_l, \tag{28}$$

and finally substituting (27) and (28) into (26) we obtain

$$g_l = c_l \cdot \sum_{j=1}^{N} f_j \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right) \tag{29}$$

for $l = 1, \ldots, N$. □

In the case when the points $\{y_j\}$ are equispaced in $[-\pi, \pi]$, the interpolation formula of Theorem 2.3 has a simpler form, which is described in the following theorem. The result of this theorem can be found in a slightly different form in [11].

Theorem 2.4 *Let $\{x_1, \ldots, x_N\}$ be a sequence of real numbers on the interval $[-\pi, \pi]$ and let $\{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ be a sequence of complex numbers. Further, let $\{y_1, \ldots, y_N\}$ be a sequence of real numbers defined by the formulae*

$$y_j = (j - 1 - N/2)\pi/N \tag{30}$$

for $j = 1, \ldots, N$, and let $\{w_1, \ldots, w_N\}$, $\{z_1, \ldots, z_N\}$, $\{f_1, \ldots, f_N\}$ and $\{g_1, \ldots, g_N\}$ be sequences of complex numbers defined by the formulae

$$w_j = e^{ix_j} \tag{31}$$

$$z_j = e^{iy_j} \tag{32}$$

$$f_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{iky_j} \tag{33}$$

$$g_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j} \tag{34}$$

for $j = 1, \ldots, N$. Then,

$$g_l = \sin\left(\frac{N x_l}{2}\right) \cdot \sum_{j=1}^{N} f_j \cdot \frac{(-1)^j}{N} \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right). \tag{35}$$

Proof. From the combination of (30) and (33), we see that the sequence $\{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ is the discrete Fourier transform of the sequence $\{f_1, \ldots, f_N\}$. In other words,

$$\alpha_k = \frac{1}{N} \cdot \sum_{j=1}^{N} f_j \cdot e^{-iky_j} \tag{36}$$

for $k = -N/2, \ldots, N/2 - 1$. Let us now define the function $f : [-\pi, \pi] \to \mathbf{C}$ by the formula

$$f(x) = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx}. \tag{37}$$

Substituting (36) into (37) and changing the order of summation, we obtain

$$
\begin{aligned}
f(x) &= \sum_{k=-N/2}^{N/2-1} \frac{1}{N} \cdot \sum_{j=1}^{N} f_j \cdot e^{-iky_j} \cdot e^{ikx} \\
&= \sum_{j=1}^{N} f_j \cdot \frac{1}{N} \cdot \sum_{k=-N/2}^{N/2-1} e^{ik(x-y_j)}.
\end{aligned} \tag{38}
$$

Observing that the second sum in the expression (38) is a geometric series we have

$$
\begin{aligned}
\sum_{k=-N/2}^{N/2-1} e^{ik(x-y_j)} &= \frac{e^{-iN(x-y_j)/2} - e^{iN(x-y_j)/2}}{1 - e^{i(x-y_j)}} \\
&= \frac{\sin(N(x-y_j)/2)}{e^{i(x-y_j)/2} \cdot \sin((x-y_j)/2)} \\
&= (\sin(N(x-y_j)/2)) \cdot (\cot((x-y_j)/2) - i)
\end{aligned} \tag{39}
$$

for any $x \in [-\pi, \pi]$. The definition of $\{y_j\}$ now yields

$$
\begin{aligned}
\sin(N(x-y_j)/2) &= \sin(Nx/2)\cos(Ny_j/2) - \cos(Nx/2)\sin(Ny_j/2) \\
&= \sin(Nx/2) \cdot (-1)^j,
\end{aligned} \tag{40}
$$

and finally, using the fact that $g_l = f(x_l)$ and combining (38), (39) and (40) we obtain

$$
g_l = \sin\left(\frac{Nx_l}{2}\right) \cdot \sum_{j=1}^{N} f_j \cdot \frac{(-1)^j}{N} \cdot \left(\frac{1}{\tan((x_l-y_j)/2)} - i\right). \tag{41}
$$

□

2.2 Relevant Facts from Approximation Theory

The algorithms of this paper are based on several results from the Chebyshev approximation theory of the function $1/\tan(x)$. These results are contained in the lemmas and theorems of this subsection, numbered 2.8–2.14. Analogs of these results for the function $1/x$ can be found in [9].

The main results of this section fall into two categories. Theorems 2.11 and 2.14 describe how the function $1/\tan(x)$ can be approximated on different regions of the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$ using Chebyshev expansions. Theorems 2.15, 2.16 and 2.17 provide three ways of manipulating these expansions which are needed by the fast algorithms of this paper.

We begin with three classical definitions which can be found, for example, in [13], [17].

Definition 2.1 *The n-th degree Chebyshev polynomial $T_n(x)$ is defined by the following equivalent formulae:*

$$
T_n(x) = \cos(n \arccos x) \tag{42}
$$

$$
T_n(x) = \frac{1}{2} \cdot \left((x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\right). \tag{43}
$$

Definition 2.2 *The roots $t_1, \ldots, t_n$ of the n-th degree Chebyshev polynomial T_n lie in the interval $[-1,1]$ and are defined by the formulae*

$$t_k = -\cos\left(\frac{2k-1}{n} \cdot \frac{\pi}{2}\right) \tag{44}$$

for $k = 1, \ldots, n$. They are referred to as Chebyshev nodes of order n.

Definition 2.3 *We will define the polynomials $u_1, \ldots, u_n$ of order $n-1$ by the formulae*

$$u_j(t) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{t - t_k}{t_j - t_k} \tag{45}$$

for $j = 1, \ldots, n$, where t_k are defined by (44).

For a function $f : [-1,1] \to \mathbf{C}$, order $n-1$ Chebyshev approximation to f on the interval $[-1,1]$ is defined as the unique polynomial of order $n-1$ which agrees with f at the nodes $t_1, \ldots, t_n$. There exist several standard representations for this polynomial, and the one we will use in this paper is given by the expression

$$\sum_{j=1}^{n} f(t_j) \cdot u_j(t). \tag{46}$$

For the purposes of this paper, Chebyshev expansions for any function will be characterized by values of this function tabulated at Chebyshev nodes.

Lemmas 2.5–2.7 provide estimates involving Chebyshev expansions which are used in the remainder of this section. The proof of Lemma 2.5 is obvious from (42).

Lemma 2.5 *Let $T_n(x)$ be the Chebyshev polynomial of degree n. Then,*

$$|T_n(x)| \leq 1 \tag{47}$$

for any $x \in [-1,1]$.

Lemma 2.6 *Let $T_n(x)$ be the Chebyshev polynomial of degree n. Then,*

$$|T_n(x)| > \frac{1}{2} \cdot \left|\frac{5x}{3}\right|^n \tag{48}$$

for any x such that $|x| \geq 3$.

Proof. From Definition 2.1, we have

$$\begin{aligned}
|T_n(x)| &= \frac{1}{2} \cdot \left|(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\right| \\
&> \frac{1}{2} \cdot |x + \sqrt{x^2 - (x/3)^2}|^n = \frac{1}{2} \cdot |x \cdot (1 + \sqrt{8/9})|^n \\
&> \frac{1}{2} \cdot \left|\frac{5x}{3}\right|^n
\end{aligned} \tag{49}$$

for any x such that $|x| \geq 3$. □

Lemma 2.7 *Let $u_j(x)$ be defined by (45). Then, for any $x \in [-1, 1]$,*

$$|u_j(x)| \leq 1. \tag{50}$$

Proof. It is obvious from (45) that $u_j(t_j) = 1$, and that $u_j(t_k) = 0$ when $k \neq j$. In addition, the expression

$$\frac{1}{n} \sum_{k=1}^{n} T_k(t_j) \cdot T_k(x) \tag{51}$$

is also equal to 1 at t_j and equal to 0 at all other t_k. Since both u_j and (51) are polynomials of order $n-1$, we have

$$u_j(x) = \frac{1}{n} \sum_{k=1}^{n} T_k(t_j) \cdot T_k(x) \tag{52}$$

for $j = 1, \ldots, n$. Furthermore, due to the combination of (52) and the triangle inequality, we obtain

$$|u_j(x)| = \left| \frac{1}{n} \sum_{k=1}^{n} T_k(t_j) \cdot T_k(x) \right| = \frac{1}{n} \sum_{k=1}^{n} |T_k(t_j)| \cdot |T_k(x)| \leq 1 \tag{53}$$

for any $x \in [-1, 1]$. □

The next lemma is obvious.

Lemma 2.8 *For any $a \in [0, \frac{\pi}{8}]$,*

$$\tan 3a \geq 3 \cdot \tan a. \tag{54}$$

The following two lemmas provide preliminary results which are used in the proof of Theorem 2.11.

Lemma 2.9 *Suppose that $n \geq 2$, and that $b > 0$ and x_0 are real numbers with $|x_0| \geq 3b$. Then, for any x,*

$$1 + xx_0 - (x - x_0) \cdot \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j\left(\frac{x}{b}\right) = (1 + x_0^2) \cdot \frac{T_n(x/b)}{T_n(x_0/b)}. \tag{55}$$

Proof. Let $Q(x)$ by the polynomial of degree n defined by the formula

$$Q(x) = 1 + xx_0 - (x - x_0) \cdot \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j\left(\frac{x}{b}\right). \tag{56}$$

It follows from the combination of (45) and (56) that

$$\begin{aligned} Q(bt_k) &= 1 + bt_k x_0 - (bt_k - x_0) \cdot \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j(t_k) \\ &= 1 + bt_k x_0 - (bt_k - x_0) \cdot \frac{1 + bt_k x_0}{bt_k - x_0} = 0, \end{aligned} \tag{57}$$

for $k = 1, \ldots, n$. Clearly, then, $Q(x)$ satisfies the conditions

$$\begin{aligned} Q(x_0) &= 1 + x_0^2 \\ Q(bt_1) &= 0 \\ &\vdots \\ Q(bt_n) &= 0. \end{aligned} \tag{58}$$

It is clear that the function

$$(1 + x_0^2) \cdot \frac{T_n(x/b)}{T_n(x_0/b)} \tag{59}$$

is also a polynomial of degree n which satisfies the $n+1$ conditions (58). Therefore,

$$Q(x) \equiv (1 + x_0^2) \cdot \frac{T_n(x/b)}{T_n(x_0/b)}, \tag{60}$$

and (55) follows as an immediate consequence of (56) and (60). □

Lemma 2.10 *Suppose that $n \geq 2$, and that $b > 0$ and x_0 are real numbers with $|x_0| \geq 3b$. Then,*

$$\left| \frac{1 + x x_0}{x - x_0} - \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j\left(\frac{x}{b}\right) \right| < \frac{1 + 9b^2}{b \cdot 5^n} \tag{61}$$

for any $x \in [-b, b]$.

Proof. Dividing (55) by $(x - x_0)$ and taking absolute values, we obtain

$$\left| \frac{1 + x x_0}{x - x_0} - \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j\left(\frac{x}{b}\right) \right| = \frac{1 + x_0^2}{|x - x_0|} \cdot \frac{|T_n(x/b)|}{|T_n(x_0/b)|}. \tag{62}$$

Due to Lemmas 2.5 and 2.6 we have

$$|T_n(x/b)| \leq 1 \tag{63}$$

for any $x \in [-b, b]$, and

$$\left| \frac{1 + x_0^2}{T_n(x_0/b)} \right| < (1 + x_0^2) \cdot 2 \cdot \left| \frac{3b}{5x_0} \right|^n < \frac{2}{5^n} \cdot (1 + (3b)^2) \tag{64}$$

for any $|x_0| \geq 3b$. Finally, substituting (63) and (64) into (62), we obtain

$$\left| \frac{1 + x x_0}{x - x_0} - \sum_{j=1}^{n} \frac{1 + bt_j x_0}{bt_j - x_0} \cdot u_j\left(\frac{x}{b}\right) \right| < \frac{1 + 9b^2}{b \cdot 5^n} \tag{65}$$

for any $x \in [-b, b]$. □

Theorem 2.11 *Suppose that $n \geq 2$, and that a and θ_0 are real numbers with $0 < a \leq \frac{\pi}{8}$ and $3a \leq |\theta_0| \leq \frac{\pi}{2}$. Then,*

$$\left| \frac{1}{\tan(\theta - \theta_0)} - \sum_{j=1}^{n} \frac{1 + t_j \tan a \tan \theta_0}{t_j \tan a - \tan \theta_0} \cdot u_j \left(\frac{\tan \theta}{\tan a} \right) \right| < \frac{1 + 9 \tan^2 a}{\tan a \cdot 5^n} \tag{66}$$

for any $\theta \in [-a, a]$.

Proof. Let $\theta \in [-a, a]$. Then, defining the real numbers b, x, and x_0 by the formulae $b = \tan a$, $x = \tan \theta$ and $x_0 = \tan \theta_0$, we observe that $|x| \leq b$, and, due to Lemma 2.8, $|x_0| \geq \tan 3a \geq 3b$. We also observe that

$$\frac{1}{\tan(\theta - \theta_0)} = \frac{1 + \tan \theta \tan \theta_0}{\tan \theta - \tan \theta_0} = \frac{1 + x x_0}{x - x_0}, \tag{67}$$

and

$$\sum_{j=1}^{n} \frac{1 + t_j \tan a \tan \theta_0}{t_j \tan a - \tan \theta_0} \cdot u_j \left(\frac{\tan \theta}{\tan a} \right) = \sum_{j=1}^{n} \frac{1 + t_j b x_0}{t_j b - x_0} \cdot u_j \left(\frac{x}{b} \right). \tag{68}$$

It follows from the combination of equations (67) and (68) and Lemma 2.10 that

$$\begin{aligned} &\left| \frac{1}{\tan(\theta - \theta_0)} - \sum_{j=1}^{n} \frac{1 + t_j \tan a \tan \theta_0}{t_j \tan a - \tan \theta_0} \cdot u_j \left(\frac{\tan \theta}{\tan a} \right) \right| \\ = &\left| \frac{1 + x x_0}{x - x_0} - \sum_{j=1}^{n} \frac{1 + t_j b x_0}{t_j b - x_0} \cdot u_j \left(\frac{x}{b} \right) \right| \\ < &\frac{1 + 9b^2}{b \cdot 5^n} = \frac{1 + 9 \tan^2 a}{\tan a \cdot 5^n} \end{aligned} \tag{69}$$

for any $\theta \in [-a, a]$. □

The following two lemmas provide preliminary results which are used in the proof of Theorem 2.14.

Lemma 2.12 *Suppose that $n \geq 2$, and that $b > 0$ and x_0 are real numbers with $|x_0| \leq b$. Then, for any x,*

$$x + 3b x_0 - (3b - x x_0) \cdot \sum_{j=1}^{n} \frac{t_j + 3b x_0}{3b - t_j x_0} \cdot u_j(x) = \left(\frac{3b}{x_0} + 3b x_0 \right) \cdot \frac{T_n(x)}{T_n(3b/x_0)}. \tag{70}$$

Proof. Let $Q(x)$ be the polynomial of degree n defined by the formula

$$Q(x) = x + 3b x_0 - (3b - x x_0) \cdot \sum_{j=1}^{n} \frac{t_j + 3b x_0}{3b - t_j x_0} \cdot u_j(x). \tag{71}$$

It follows from the combination of (45) and (71) that

$$\begin{aligned} Q(t_k) &= t_k + 3bx_0 - (3b - t_k x_0) \cdot \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(t_k) \\ &= t_k + 3bx_0 - (3b - t_k x_0) \cdot \frac{t_k + 3bx_0}{3b - t_k x_0} = 0, \end{aligned} \tag{72}$$

for $k = 1, \ldots, n$. Clearly, then, $Q(x)$ satisfies the conditions

$$\begin{aligned} Q(3b/x_0) &= 3b/x_0 + 3bx_0 \\ Q(t_1) &= 0 \\ &\vdots \\ Q(t_n) &= 0. \end{aligned} \tag{73}$$

It is clear that the function

$$\left(\frac{3b}{x_0} + 3bx_0\right) \cdot \frac{T_n(x)}{T_n(3b/x_0)} \tag{74}$$

is also a polynomial of degree n which satisfies the $n+1$ conditions (73). Therefore,

$$Q(x) \equiv \left(\frac{3b}{x_0} + 3bx_0\right) \cdot \frac{T_n(x)}{T_n(3b/x_0)}, \tag{75}$$

and (70) follows as an immediate consequence of (71) and (75). □

Lemma 2.13 *Suppose that $n \geq 2$, and that $b > 0$ and x_0 are real numbers with $|x_0| \leq b$. Then,*

$$\left| \frac{x + 3bx_0}{3b - xx_0} - \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(x) \right| < \frac{3(1+b^2)}{b \cdot 5^n} \tag{76}$$

for any $x \in [-1, 1]$.

Proof. Dividing (70) by $(3b - xx_0)$ and taking absolute values, we obtain

$$\left| \frac{x + 3bx_0}{3b - xx_0} - \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(x) \right| = \frac{1}{|3b - xx_0|} \cdot \left| \frac{3b}{x_0} + 3bx_0 \right| \cdot \frac{|T_n(x)|}{|T_n(3b/x_0)|}. \tag{77}$$

In addition, due to Lemmas 2.5 and 2.6 we have

$$|T_n(x)| \leq 1 \tag{78}$$

for any $x \in [-1, 1]$, and

$$\left| \frac{3b/x_0 + 3bx_0}{T_n(3b/x_0)} \right| < 3b \cdot (1/x_0 + x_0) \cdot 2 \cdot \left| \frac{3x_0}{5 \cdot 3b} \right|^n < \frac{6b}{5^n} \cdot (1/b + b) \tag{79}$$

for $|x_0| \leq b$. Substituting (78) and (79) into (77), we obtain

$$\left| \frac{x + 3bx_0}{3b - xx_0} - \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(x) \right| \leq \frac{1}{2b} \cdot \frac{6b}{5^n} \cdot (1/b + b) = \frac{3(1+b^2)}{b \cdot 5^n} \tag{80}$$

for any $x \in [-1, 1]$. □

Theorem 2.14 *Suppose that $n \geq 2$, and that a and θ_0 are real numbers with $0 < a \leq \frac{\pi}{8}$ and $|\theta_0| \leq a$. Then,*

$$\left| \frac{1}{\tan(\theta - \theta_0)} - \sum_{j=1}^{n} \frac{t_j + 3\tan a \tan\theta_0}{3\tan a - t_j \tan\theta_0} \cdot u_j\left(\frac{3\tan a}{\tan\theta}\right) \right| < \frac{6}{\sin 2a \cdot 5^n} \tag{81}$$

for any θ such that $3a \leq |\theta| \leq \frac{\pi}{2}$.

Proof. Let θ be any real number such that $3a \leq |\theta| \leq \frac{\pi}{2}$. Then, defining the real numbers b, x and x_0 by the formulae $b = \tan a$, $x = 3\tan a/\tan\theta$ and $x_0 = \tan\theta_0$, we observe that $|x_0| \leq b$, and, due to Lemma 2.8, $|x| \leq 1$. We also observe that

$$\frac{1}{\tan(\theta - \theta_0)} = \frac{1 + \tan\theta\tan\theta_0}{\tan\theta - \tan\theta_0} = \frac{1 + 3bx_0/x}{3b/x - x_0} = \frac{x + 3bx_0}{3b - xx_0}, \tag{82}$$

and

$$\sum_{j=1}^{n} \frac{t_j + 3\tan a \tan\theta_0}{3\tan a - t_j \tan\theta_0} \cdot u_j\left(\frac{3\tan a}{\tan\theta}\right) = \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(x). \tag{83}$$

It follows from the combination of equations (82) and (82) and Lemma 2.12 that

$$\begin{aligned} &\left| \frac{1}{\tan(\theta - \theta_0)} - \sum_{j=1}^{n} \frac{t_j + 3\tan a \tan\theta_0}{3\tan a - t_j \tan\theta_0} \cdot u_j\left(\frac{3\tan a}{\tan\theta}\right) \right| \\ &= \left| \frac{x + 3bx_0}{3b - xx_0} - \sum_{j=1}^{n} \frac{t_j + 3bx_0}{3b - t_j x_0} \cdot u_j(x) \right| && (84) \\ &< 3 \cdot \frac{1 + b^2}{b \cdot 5^n} = \frac{3\sec^2 a}{\tan a \cdot 5^n} = \frac{6}{\sin 2a \cdot 5^n} && (85) \end{aligned}$$

for any θ such that $3a \leq |\theta| \leq \frac{\pi}{2}$. □

The following three theorems provide formulae for translating along the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$ Chebyshev expansions of the type described in the previous two theorems. Theorem 2.15 provides a formula for translating expansions described in Theorems 2.11, Theorem 2.16 describes a mechanism of converting the expansion of Theorem 2.14 to the expansion of Theorem 2.11, and Theorem 2.17 provides a way of translating the expansion of Theorem 2.14.

Theorem 2.15 *Suppose that $n, N \geq 2$, and let a, c, d be real numbers such that $0 < a \leq \pi/8$ and $[c-d, c+d] \subset [-a, a]$. Let the function $f : [-\frac{\pi}{2}, \frac{\pi}{2}] \to \mathbf{C}$ be defined by the formula*

$$f(\theta) = \sum_{k=1}^{N} \frac{\alpha_k}{\tan(\theta - \theta_k)} \tag{86}$$

where $3a \leq |\theta_k| \leq \frac{\pi}{2}$ for $k = 1, \ldots, N$, and $\alpha_1, \ldots, \alpha_N$ is a set of complex numbers. Further, let $\Psi_1, \ldots, \Psi_n$ be a set of complex numbers defined by the formula

$$\Psi_k = f(\arctan(t_k \tan a)) \tag{87}$$

for $k = 1, \ldots, n$, and let $\tilde{\Psi}_1, \ldots, \tilde{\Psi}_n$ be a set of complex numbers defined by the formula

$$\tilde{\Psi}_k = \sum_{j=1}^{n} \Psi_j \cdot u_j \left(\frac{\tan(c + \arctan(t_k \tan d))}{\tan a} \right) \tag{88}$$

for $k = 1, \ldots, n$. Then, for any $\theta \in [c-d, c+d]$,

$$\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Psi}_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| < A \cdot \frac{(n+1)(1 + 9\tan^2 a)}{\tan a \cdot 5^n}, \tag{89}$$

where $A = \sum_{k=1}^{N} |\alpha_k|$.

Proof. It follows from the triangle inequality that

$$\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Psi}_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| \leq S_1 + S_2 \tag{90}$$

where

$$S_1 = \left| f(\theta) - \sum_{k=1}^{n} f(c + \arctan(t_k \tan d)) \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right|, \tag{91}$$

and

$$S_2 = \left| \sum_{k=1}^{n} (f(c + \arctan(t_k \tan d)) - \tilde{\Psi}_k) \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right|. \tag{92}$$

Combining Theorem 2.11 with Lemma 2.7 and the triangle inequality, we have

$$S_1 < A \cdot \frac{1 + 9\tan^2 a}{\tan a \cdot 5^n}, \tag{93}$$

and

$$\begin{aligned} S_2 &\leq \sum_{k=1}^{n} \left| f(c + \arctan(t_k \tan d)) - \sum_{j=1}^{n} \Psi_j \cdot u_j \left(\frac{\tan(c + \arctan(t_k \tan d))}{\tan a} \right) \right| \\ &< An \cdot \frac{1 + 9\tan^2 a}{\tan a \cdot 5^n}, \end{aligned} \tag{94}$$

where $A = \sum_{k=1}^{N} |\alpha_k|$. Finally, substituting (93) and (94) into (90) we obtain

$$\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Psi}_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| < A \cdot (n+1) \cdot \frac{1 + 9\tan^2 a}{\tan a \cdot 5^n} \tag{95}$$

for any $\theta \in [c-d, c+d]$. □

Theorem 2.16 *Suppose that $n, N \geq 2$, and let a, c, d be real numbers such that $0 < a \leq \pi/8$ and $|c| - d > 3a$. Let the function $f : [-\frac{\pi}{2}, \frac{\pi}{2}] \to \mathbf{C}$ be defined by the formula*

$$f(\theta) = \sum_{k=1}^{N} \frac{\alpha_k}{\tan(\theta - \theta_k)} \tag{96}$$

where $\theta_k \in [-a, a]$ for $k = 1, \ldots, N$, and $\alpha_1, \ldots, \alpha_N$ is a set of complex numbers. Further, let $\Phi_1, \ldots, \Phi_n$ be a set of complex numbers defined by the formula

$$\Phi_k = f(\arctan(3\tan(a)/t_k)) \tag{97}$$

for $k = 1, \ldots, n$, and let $\Psi_1, \ldots, \Psi_n$ be a set of complex numbers defined by the formula

$$\Psi_k = \sum_{j=1}^{n} \Phi_j \cdot u_j \left(\frac{3\tan a}{\tan(c + \arctan(t_k \tan d))} \right) \tag{98}$$

for $k = 1, \ldots, n$. Then, for any $\theta \in [c-d, c+d]$,

$$\left| f(\theta) - \sum_{k=1}^{n} \Psi_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| < A \cdot \frac{3n\sec^2 a + 1 + 9\tan^2 a}{\tan a \cdot 5^n} \tag{99}$$

where $A = \sum_{k=1}^{N} |\alpha_k|$.

Proof. It follows from the triangle inequality that

$$\left| f(\theta) - \sum_{k=1}^{n} \Psi_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| \leq S_1 + S_2 \tag{100}$$

where

$$S_1 = \left| f(\theta) - \sum_{k=1}^{n} f(c + \arctan(t_k \tan d)) \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right|, \tag{101}$$

and

$$S_2 = \left| \sum_{k=1}^{n} (f(c + \arctan(t_k \tan d)) - \Psi_k) \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right|. \tag{102}$$

Combining Theorem 2.11 with the triangle inequality gives us

$$S_1 < A \cdot \frac{1 + 9\tan^2 a}{\tan a \cdot 5^n}, \tag{103}$$

and from the combination of Theorem 2.14, Lemma 2.7 and the triangle inequality, we have

$$
\begin{aligned}
S_2 &\le \sum_{k=1}^{n} \left| f(c+\arctan(t_k \tan d)) - \sum_{j=1}^{n} \Phi_j \cdot u_j \left(\frac{3\tan a}{\tan(c+\arctan(t_k \tan d))} \right) \right| \\
&< An \cdot \frac{3\sec^2 a}{\tan a \cdot 5^n},
\end{aligned} \tag{104}
$$

where $A = \sum_{k=1}^{N} |\alpha_k|$. Finally, substituting (103) and (104) into (100) we obtain

$$
\left| f(\theta) - \sum_{k=1}^{n} \Psi_k \cdot u_j \left(\frac{\tan(\theta - c)}{\tan d} \right) \right| < A \cdot \frac{3n\sec^2 a + 1 + 9\tan^2 a}{\tan a \cdot 5^n} \tag{105}
$$

for any $\theta \in [c-d, c+d]$. □

Theorem 2.17 *Suppose that $n, N \ge 2$, and let a, c, d be real numbers such that $0 < a \le \pi/8$ and $[c-d, c+d] \supset [-a, a]$. Let the function $f : [-\frac{\pi}{2}, \frac{\pi}{2}] \to \mathbf{C}$ be defined by the formula*

$$
f(\theta) = \sum_{k=1}^{N} \frac{\alpha_k}{\tan(\theta - \theta_k)} \tag{106}
$$

where $\theta_k \in [-a, a]$ for $k = 1, \ldots, N$, and $\alpha_1, \ldots, \alpha_N$ is a set of complex numbers. Further, let $\Phi_1, \ldots, \Phi_n$ be a set of complex numbers defined by the formula

$$
\Phi_k = f(\arctan(3\tan(a)/t_k)) \tag{107}
$$

for $k = 1, \ldots, n$, and let $\tilde{\Phi}_1, \ldots, \tilde{\Phi}_n$ be a set of complex numbers defined by the formula

$$
\tilde{\Phi}_k = \sum_{j=1}^{n} \Phi_j \cdot u_j \left(\frac{3\tan a}{\tan(c + \arctan(3\tan(d)/t_k))} \right) \tag{108}
$$

for $k = 1, \ldots, n$. Then, for any θ such that $|\theta - c| \ge 3d$,

$$
\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Phi}_k \cdot u_j \left(\frac{3\tan d}{\tan(\theta - c)} \right) \right| < A \cdot \frac{3(n+1)\sec^2 a}{\tan a \cdot 5^n}, \tag{109}
$$

where $A = \sum_{k=1}^{N} |\alpha_k|$.

Proof. It follows from the triangle inequality that

$$
\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Phi}_k \cdot u_j \left(\frac{3\tan d}{\tan(\theta - c)} \right) \right| \le S_1 + S_2 \tag{110}
$$

where

$$
S_1 = \left| f(\theta) - \sum_{k=1}^{n} f(c + \arctan(3\tan(d)/t_k)) \cdot u_j \left(\frac{3\tan d}{\tan(\theta - c)} \right) \right|, \tag{111}
$$

and

$$S_2 = \left|\sum_{k=1}^{n} \left(f(c + \arctan(3\tan(d)/t_k)) - \tilde{\Phi}_k\right) \cdot u_j\left(\frac{3\tan d}{\tan(\theta - c)}\right)\right|. \tag{112}$$

Combining Theorem 2.14 with Lemma 2.7 and the triangle inequality, we have

$$S_1 < A \cdot \frac{3\sec^2 a}{\tan a \cdot 5^n}, \tag{113}$$

and

$$\begin{aligned} S_2 &\le \sum_{k=1}^{n} \left| f(c + \arctan(3\tan(d)/t_k)) - \sum_{j=1}^{n} \Phi_j \cdot u_j\left(\frac{3\tan a}{\tan(c + \arctan(3\tan(d)/t_k))}\right)\right| \\ &< An \cdot \frac{3\sec^2 a}{\tan a \cdot 5^n}, \end{aligned} \tag{114}$$

where $A = \sum_{k=1}^{N} |\alpha_k|$. Finally, substituting (113) and (114) into (110) we obtain

$$\left| f(\theta) - \sum_{k=1}^{n} \tilde{\Phi}_k \cdot u_j\left(\frac{3\tan d}{\tan(\theta - c)}\right)\right| < A \cdot \frac{3(n+1)\sec^2 a}{\tan a \cdot 5^n}, \tag{115}$$

for any θ such that $|\theta - c| \ge 3d$. $\square$

3 Application of the FMM to Nonequispaced FFTs

For the remainder of this paper we will be considering the mapping $F : \mathbf{C}^N \to \mathbf{C}^N$ defined by the formulae

$$F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j}, \tag{116}$$

for $j = 1, \ldots, N$, where $x = \{x_1, \ldots, x_N\}$ is a sequence of real numbers in $[-\pi, \pi]$ and $\alpha = \{\alpha_1, \ldots, \alpha_N\}$ is a sequence of complex numbers. We are interested in the efficient application and inversion of the transformation F and its transpose. More formally, we will consider the following four problems:

- **Problem 1:** Given α, find $f = F(\alpha)$.
- **Problem 2:** Given α, find $f = F^T(\alpha)$.
- **Problem 3:** Given f, find $\alpha = F^{-1}(f)$.
- **Problem 4:** Given f, find $\alpha = (F^T)^{-1}(f)$.

This section consists of four parts. In Subsection 3.1 we describe briefly how the one dimensional Fast Multipole algorithm of [9] can be applied to the problems of this paper, in Subsection 3.2 we outline a set of four algorithms for these problems, Subsection 3.3 contains more formal descriptions of these algorithms, and finally in Subsection 3.4 we discuss a generalization of Problems 1–4.

3.1 FMM and Trigonometric Interpolation

There exist a number of different formulations of the trigonometric interpolation problem (see [17]). The version we will use for the purposes of this paper is described as follows: given a set of points $\{y_1, \ldots, y_N\}$ and function values $\{f_1, \ldots, f_N\}$, evaluate the interpolating Fourier series at the points $\{x_1, \ldots, x_N\}$. According to Theorem 2.3, these values are given by the formulae

$$g_l = c_l \cdot \sum_{j=1}^{N} f_j \cdot d_j \cdot \left(\frac{1}{\tan((x_l - y_j)/2)} - i \right) \tag{117}$$

for $l = 1, \ldots, N$, where $\{c_l\}$ and $\{d_j\}$ are defined by the formulae

$$c_l = \prod_{k=1}^{N} \sin((x_l - y_k)/2) = e^{\sum_{k=1}^{N} \ln(\sin((x_l - y_k)/2))} \tag{118}$$

for $l = 1, \ldots, N$, and

$$d_j = \prod_{\substack{k=1 \\ k \neq j}}^{N} \frac{1}{\sin((y_j - y_k)/2)} = e^{-\sum_{k=1, k\neq j}^{N} \ln(\sin((y_j - y_k)/2))} \tag{119}$$

for $j = 1, \ldots, N$.

Remark 3.1 The FMM algorithms of [9] are designed to evaluate expressions of the form

$$\sum_{k=1}^{N} \frac{\alpha_k}{x - x_k} \tag{120}$$

in $O\left(N \log\left(\frac{1}{\varepsilon}\right)\right)$ arithmetic operations, where ε is the desired accuracy. With a few minor modifications they can also be used to evaluate expressions of the form

$$\sum_{k=1}^{N} \frac{\alpha_k}{\tan(x - x_k)} \tag{121}$$

and

$$\sum_{k=1}^{N} \ln(\sin(x - x_k)), \tag{122}$$

and hence expressions of the form (117) for the same computational cost. Moreover, the algorithmic procedure for the kernel $1/\tan x$ is virtually identical to that for $1/x$, and the various expansions required by the algorithms are manipulated via Theorems 2.15, 2.16 and 2.17 (see [9] for detailed descriptions of these algorithms).

3.2 Informal Descriptions of the Algorithms

In this subsection we outline how a fast trigonometric interpolation scheme can be used to construct efficient algorithms for Problems 1–4 of this paper.

We begin with some notation.

$\mathcal{F} : \mathbf{C}^N \to \mathbf{C}^N$ will denote the matrix which maps a sequence of N complex numbers to its discrete Fourier transform. $\mathcal{F}$ is defined by the formulae

$$\mathcal{F}_{jk} = e^{2\pi i\cdot(j-N/2-1)\cdot(k-N/2-1)/N} \tag{123}$$

for $j,k = 1,\ldots,N$, and it is well known that $\mathcal{F}^T = \mathcal{F}$, and that $\mathcal{F}^{-1} = \frac{1}{N}\bar{\mathcal{F}}$.

Remark 3.2 $\mathcal{F}$ and $\mathcal{F}^{-1}$ can each be applied in $O(N \log N)$ operations via the FFT.

$\mathcal{P} : \mathbf{C}^N \to \mathbf{C}^N$ will denote the matrix which maps the values of an N-term Fourier series at N equispaced points $\{y_1,\ldots,y_N\}$ on $[-\pi,\pi]$ to the values of this series at the arbitrarily spaced points $\{x_1,\ldots,x_N\}$. According to Theorem 2.4, $\mathcal{P}$ is defined by the formulae

$$\mathcal{P}_{jk} = \sin\left(\frac{Nx_j}{2}\right)\cdot\frac{(-1)^k}{N}\cdot\left(\frac{1}{\tan((x_j-y_k)/2)} - i\right) \tag{124}$$

for $j,k = 1,\ldots,N$. It follows directly from (124) that

$$\mathcal{P}^T_{jk} = \frac{(-1)^j}{N}\cdot\sin\left(\frac{Nx_k}{2}\right)\cdot\left(\frac{1}{\tan((x_k-y_j)/2)} - i\right) \tag{125}$$

for $j,k = 1,\ldots,N$. The inverse of the mapping $\mathcal{P}$ converts the values of an N-term Fourier series at the points $\{x_1,\ldots,x_N\}$ to the values of this series at the equispaced points $\{y_1,\ldots,y_N\}$. $\mathcal{P}^{-1}$ is therefore given analytically, and according to Theorem 2.4 it is defined by the formulae

$$\mathcal{P}^{-1}_{jk} = c_j\cdot d_k\cdot\left(\frac{1}{\tan((y_j-x_k)/2)} - i\right) \tag{126}$$

for $j,k = 1,\ldots,N$, where $c_1,\ldots,c_N$ and $d_1,\ldots,d_N$ are sequences of real numbers defined by the formulae (118) and (119). It follows directly from (126) that

$$(\mathcal{P}^T)^{-1}_{jk} = d_j\cdot c_k\cdot\left(\frac{1}{\tan((y_k-x_j)/2)} - i\right) \tag{127}$$

for $j,k = 1,\ldots,N$.

Remark 3.3 $\mathcal{P}$, $\mathcal{P}^T$, $\mathcal{P}^{-1}$ and $(\mathcal{P}^T)^{-1}$ are all of the same form, and each can be applied with a relative precision ε in $O\left(N\log\left(\frac{1}{\varepsilon}\right)\right)$ operations via the FMM (see Section 3.1).

Observation 3.4 *From the combination of (116), Theorems 2.3 and 2.4, and several elementary matrix identities, we see that*

$$\begin{aligned} F &= \mathcal{P} \cdot \mathcal{F} \\ F^T &= \mathcal{F} \cdot \mathcal{P}^T \\ F^{-1} &= \mathcal{F}^{-1} \cdot \mathcal{P}^{-1} \\ (F^T)^{-1} &= (\mathcal{P}^T)^{-1} \cdot \mathcal{F}^{-1}. \end{aligned} \tag{128}$$

Furthermore, due to Remarks 3.2 and 3.3, F, F^T, F^{-1} and $(F^T)^{-1}$ can each be applied in $O\left(N \log\left(\frac{1}{\varepsilon}\right)\right)$ arithmetic operations.

3.3 Formal Descriptions of the Algorithms

Following are detailed descriptions of the four algorithms of this paper.

Algorithm 1

Step	**Complexity**	**Description**
1	$O(N \log N)$	**Comment** [Evaluate Fourier series at equispaced points using FFT.] Compute $g_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{iky_j}$ for $j = 1, \ldots, N$.

2	$O(N \log(\frac{1}{\varepsilon}))$	**Comment** [Interpolate in space domain.]
		do $j = 1, N$
		    $g_j = g_j \cdot (-1)^j / N$
		end do
		Compute $f_l = \sum_{j=1}^{N} g_j / \tan((x_l - y_j)/2)$ for $l = 1, \ldots, N$ using FMM.
		do $l = 1, N$
		    $f_l = f_l - i \cdot \sum_{j=1}^{N} g_j$
		end do
		do $l = 1, N$
		    $f_l = f_l \cdot \sin(N x_l / 2)$
		end do

Total $O(N \cdot \log N + N \cdot \log(\frac{1}{\varepsilon}))$

Algorithm 2

Step	**Complexity**	**Description**

1	$O(N \log(\frac{1}{\varepsilon}))$	**Comment** [Interpolate in frequency domain.]
		do $j = 1, N$
		    $\alpha_j = \alpha_j \cdot \sin(N x_j / 2)$
		end do
		Compute $a_l = -\sum_{j=1}^{N} \alpha_j / \tan((y_l - x_j)/2)$ for $l = 1, \ldots, N$ using FMM.
		do $l = 1, N$
		    $a_l = a_l - i \cdot \sum_{j=1}^{N} \alpha_j$
		end do
		do $l = 1, N$
		    $a_l = a_l \cdot (-1)^l / N$
		end do

2	$O(N \log N)$	**Comment** [Evaluate Fourier series at equispaced points using FFT.]
		Compute $f_j = \sum_{k=-N/2}^{N/2-1} a_k \cdot e^{iky_j}$ for $j = 1, \ldots, N$.

Total $O(N \cdot \log N + N \cdot \log(\frac{1}{\epsilon}))$

Algorithm 3

Step	Complexity	Description
1	$O(N \log(\frac{1}{\epsilon}))$	**Comment** [Interpolate in space domain.]

do $j = 1, N$
  $f_j = f_j \cdot d_j$
end do
Compute $a_l = \sum_{j=1}^{N} f_j / \tan((y_l - x_j)/2)$ for $l = 1, \ldots, N$ using FMM.
do $l = 1, N$
  $a_l = a_l - i \cdot \sum_{j=1}^{N} f_j$
end do
do $l = 1, N$
  $a_l = a_l \cdot c_l$
end do

2	$O(N \log N)$	**Comment** [Obtain Fourier coeffients using FFT.]
		Compute $\alpha_j = \frac{1}{N} \cdot \sum_{k=1}^{N} a_k \cdot e^{-iky_j}$ for $j = -N/2, \ldots, N/2 - 1$.

Total $O(N \cdot \log N + N \cdot \log(\frac{1}{\epsilon}))$

Algorithm 4

Step	Complexity	Description
1	$O(N \log N)$	**Comment** [Obtain Fourier coeffients using FFT.]
		Compute $a_j = \frac{1}{N} \cdot \sum_{k=1}^{N} f_k \cdot e^{-iky_j}$ for $j = 1, \ldots, N$.
2	$O(N \log(\frac{1}{\epsilon}))$	**Comment** [Interpolate in frequency domain.]

do $j = 1, N$
  $a_j = a_j \cdot c_j$
end do
Compute $\alpha_l = -\sum_{j=1}^{N} a_j / \tan((x_l - y_j)/2)$ for $l = 1, \ldots, N$ using FMM.
do $l = 1, N$
  $\alpha_l = \alpha_l - i \cdot \sum_{j=1}^{N} a_j$
end do
do $l = 1, N$
  $\alpha_l = \alpha_l \cdot d_l$
end do

Total $O(N \cdot \log N + N \cdot \log(\frac{1}{\epsilon}))$

3.4 FFTs for Complex Data Points

Various generalizations of the problems addressed in this paper are mentioned briefly in Section 5. One of the generalizations of Problems 1–4 merits special attention, and is discussed in this section: this is the case when the points $\{x_j\}$ are complex, and lie slightly off the real axis.

We are interested here in the transformations described by the formulae

$$f_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikr_j} \cdot e^{-ks_j}, \tag{129}$$

for $j = 1, \ldots, N$, which is a generalization of (116) with

$$x_j = r_j + is_j. \tag{130}$$

Algorithms 1–4 can be modified to evaluate expressions of the form (129), provided that the s_j are small (on the order of $\frac{1}{N}$).

Problems of this type are frequently encountered in signal analysis, computational complex analysis and several other areas.

4 Implementation and Numerical Results

We have written FORTRAN implementations of the algorithms of this paper using double precision arithmetic, and have applied these programs to a variety of situations. This section contains results of four of our numerical experiments demonstrating the performance of our implementations of Algorithms 1–4.

Several technical details of our implementations appear to be worth mentioning here:

1. Each implementation consists of two main subroutines: the first is an initialization stage in which the elements of the various matrices employed by the algorithms are precomputed and stored, and the second is an evaluation stage in which these matrices are applied. Successive application of the linear transformations to multiple vectors requires the initialization to be performed only once.

2. The algorithms of this paper all require the evaluation of sums of the form

$$f_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{2\pi ikj/N} \tag{131}$$

for $j = -N/2, \ldots, N/2-1$, whereas most FFT software computes sums of the form

$$f_j = \sum_{k=0}^{N-1} \alpha_k \cdot e^{2\pi ikj/N} \tag{132}$$

for $j = 0, \ldots, N-1$. We used a standard FFT to evaluate sums of the form (131) by defining $\hat{\alpha}_k = \alpha_k$ for $k = 0, \ldots, N/2-1$, $\hat{\alpha}_k = \alpha_{k-N}$ for $k = N/2, \ldots, N-1$, $\hat{f}_j = f_j$ for $j = 0, \ldots, N/2-1$ and $\hat{f}_j = f_{k-N}$ for $j = N/2, \ldots, N-1$. This substitution converts the form (131) to the form (132).

3. The algorithms of this paper require an FFT of size proportional to N, and will thus perform efficiently whenever the FFT does. This restriction on problem size can be removed by extending the input vector to length $2^{\lceil \log_2 N \rceil}$ (i.e. the smallest power of 2 which is greater than N) and padding it with zeroes. This ensures that the algorithms will perform efficiently for any choice of N. In our implementations these changes were made.

4. Each of the algorithms described in Section 3 utilizes a version of the one dimensional FMM. The version used in our implementations was Algorithm 3.2 of [9] which was chosen to maximize both efficiency and accuracy.

Our implementations of the algorithms of this paper have been tested on the the Sun SPARCstation 1 for a variety of input data. Four experiments are described in this section and their results are summarized in Tables 1–4. These tables contain error estimates and CPU time requirements for the algorithms, with all computations performed in double precision arithmetic.

The table entries are described below.

- The first column in each table contains the problem size N, which was chosen to be a power of 2 ranging from 128 to 2048 for each example.

- The second column in each table contains the relative ∞-norm error defined by the formula

$$E_\infty = \max_{1\le j\le N} |\tilde{f}_j - f_j| \Big/ \max_{1\le j\le N} |f_j| , \tag{133}$$

  where the vector $\tilde{f}$ is the algorithm output and the vector f is the result of a direct calculation.

- The third column in each table contains the relative 2-norm error defined by the formula

$$E_2 = \sqrt{\sum_{j=1}^{N} |\tilde{f}_j - f_j|^2 \Big/ \sum_{j=1}^{N} |f_j|^2}, \tag{134}$$

  where the vector $\tilde{f}$ is the algorithm output and the vector f is the result of a direct calculation.

- The fourth and fifth columns in each table contain CPU timings for the initialization and evaluation stages of the algorithm.

- The sixth column in each table contains CPU timings for the corresponding direct calculation.

- The last column in Tables 1 and 2 contains CPU timings for an FFT of the same size.

Remark 4.1 Our implementations of the direct methods for Examples 1 and 2 were optimized by using the fact that $e^{ikx_j} = (e^{ix_j})^k$ to reduce the number of complex exponential computations.

Remark 4.2 Standard LINPACK Gaussian Elimination subroutines were used as the direct methods for comparing timings in Examples 3 and 4. Estimated timings are presented for larger N, where this computation became impractical.

Following are the descriptions of the experiments, and the tables of numerical results.

Example 1.

Here we consider the transformation $F : \mathbf{C}^N \to \mathbf{C}^N$ of Problem 1, defined by the formula

$$F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j} \tag{135}$$

for $j = 1, \ldots, N$. In this example, $\{x_1, \ldots, x_N\}$ were randomly distributed on the interval $[-\pi, \pi]$ and $\{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ were complex numbers randomly chosen from the unit square

$$0 \le \mathrm{Re}(z) \le 1,\ 0 \le \mathrm{Im}(z) \le 1. \tag{136}$$

The results of applying Algorithm 1 to this problem are presented in Table 1.

Example 2.

Here we consider the transformation $F^T : \mathbf{C}^N \to \mathbf{C}^N$ of Problem 2, defined by the formula

$$F^T(\alpha)_j = \sum_{k=1}^{N} \alpha_k \cdot e^{ijx_k} \tag{137}$$

for $j = -N/2, \ldots, N/2 - 1$. In this example, $\{x_1, \ldots, x_N\}$ were randomly distributed on the interval $[-\pi, \pi]$ and $\{\alpha_1, \ldots, \alpha_N\}$ were complex numbers randomly chosen from the unit square

$$0 \le \mathrm{Re}(z) \le 1,\ 0 \le \mathrm{Im}(z) \le 1. \tag{138}$$

The results of applying Algorithm 2 to this problem are presented in Table 2.

Example 3.

Here we consider the transformation $F^{-1} : \mathbf{C}^N \to \mathbf{C}^N$ of Problem 3 where F is defined by the formula

$$F(\alpha)_j = \sum_{k=-N/2}^{N/2-1} \alpha_k \cdot e^{ikx_j} \tag{139}$$

for $j = 1, \ldots, N$. In this example, $\{x_1, \ldots, x_N\}$ were defined by the formulae

$$x_j = -\pi + 2\pi \cdot \frac{j + 0.5 + \delta_j}{N} \tag{140}$$

for $j = 1, \ldots, N$, where δ_j were randomly distributed on the interval $[-0.1, 0.1]$. In addition, $\{\alpha_{-N/2}, \ldots, \alpha_{N/2-1}\}$ were complex numbers randomly chosen from the unit square

$$0 \le \mathrm{Re}(z) \le 1,\ 0 \le \mathrm{Im}(z) \le 1, \tag{141}$$

and the numbers $\{f_1, \ldots, f_N\}$ were computed directly in double precision arithmetic according to the formula $f_j = F(\alpha)_j$. The vector f was then used as input for Algorithm 3. Results of this experiment are presented in Table 3.

Example 4.
Here we consider the transformation $(F^T)^{-1} : \mathbf{C}^N \rightarrow \mathbf{C}^N$ of Problem 4 where F^T is defined by the formula

$$F^T(\alpha)_j = \sum_{k=1}^{N} \alpha_k \cdot e^{ijx_k} \tag{142}$$

for $j = -N/2, \ldots, N/2 - 1$. In this example, $\{x_1, \ldots, x_N\}$ were defined by the formulae

$$x_j = -\pi + 2\pi \cdot \frac{j + 0.5 + \delta_j}{N} \tag{143}$$

for $j = 1, \ldots, N$, where δ_j were randomly distributed on the interval $[-0.1, 0.1]$. In addition, $\{\alpha_1, \ldots, \alpha_N\}$ were complex numbers randomly chosen from the unit square

$$0 \leq \mathrm{Re}(z) \leq 1,\ 0 \leq \mathrm{Im}(z) \leq 1, \tag{144}$$

and the numbers $\{f_{-N/2}, \ldots, f_{N/2-1}\}$ were computed directly in double precision arithmetic according to the formula $f_j = F^T(\alpha)_j$. The vector f was then used as input for Algorithm 4. Results of this experiment are presented in Table 4.

The following observations can be made from Tables 1–4, and are in agreement with results of our more extensive experiments for this particular computer architecture, implementation and range of N.

1. All of the algorithms permit high accuracy to be attained, and the observed errors are in accordance with the theoretically obtained error bounds.
2. As expected, the CPU timings for all the algorithms grow slightly faster than linearly with the problem size N.
3. The timings for Algorithms 1–4 are similar, which is to be expected since these four algorithms are so closely related.
4. The initialization times for Algorithms 1 and 2 are less than those for Algorithms 3 and 4. This is because the former pair does not incur the additional cost of computing the numbers $\{c_k\}$ and $\{d_k\}$.
5. The evaluation stages of Algorithms 1–4 are about 15 times as costly as an FFT of the same size.
6. Algorithms 1 and 2 can compete with the direct method at about $N = 32$ ignoring initialization time, and at $N = 1024$ including the initialization. Algorithms 3 and 4 are always dramatically faster than the direct calculation (15000 times faster at $N = 2048$) if we ignore initialization time, and break even with it at $N = 64$ if we include the initialization.

N	Errors		Timings (sec.)			
	E_∞	E_2	Alg. Init.	Alg. Eval.	Direct	FFT
128	0.379 E-14	0.704 E-14	1.04	0.030	0.09	0.002
256	0.398 E-14	0.116 E-13	2.03	0.081	0.33	0.005
512	0.499 E-14	0.195 E-13	3.26	0.171	1.24	0.012
1024	0.318 E-13	0.625 E-13	4.97	0.408	4.93	0.026
2048	0.763 E-13	0.204 E-12	8.07	0.822	19.62	0.059

Table 1: Example 1, Numerical Results for Algorithm 1.

N	Errors		Timings (sec.)			
	E_∞	E_2	Alg. Init.	Alg. Eval.	Direct	FFT
128	0.206 E-14	0.800 E-14	1.03	0.033	0.08	0.002
256	0.323 E-14	0.136 E-13	2.05	0.081	0.31	0.005
512	0.153 E-13	0.343 E-13	3.21	0.174	1.20	0.012
1024	0.180 E-13	0.654 E-13	5.11	0.409	4.76	0.026
2048	0.470 E-13	0.221 E-12	8.16	0.823	18.93	0.059

Table 2: Example 2, Numerical Results for Algorithm 2.

7. The initialization stage is much more costly than the evaluation stage for all of the algorithms. Implementing the algorithms in two stages thus gives considerable time savings whenever the same linear transformation is to be applied to multiple vectors.

N	Errors		Timings (sec.)			
	E_∞	E_2	Alg. Init.	Alg. Eval.	Direct	FFT
128	0.117 E-13	0.800 E-14	1.28	0.034	2.96	0.002
256	0.196 E-13	0.137 E-13	2.51	0.082	23.6	0.005
512	0.344 E-13	0.230 E-13	4.33	0.175	189	0.012
1024	0.107 E-12	0.757 E-13	7.45	0.409	1512 (est.)	0.026
2048	0.357 E-12	0.247 E-12	12.97	0.819	12096 (est.)	0.059

Table 3: Example 3, Numerical Results for Algorithm 3.

N	Errors		Timings (sec.)			
	E_∞	E_2	Alg. Init.	Alg. Eval.	Direct	FFT
128	0.134 E-13	0.806 E-14	1.26	0.033	2.96	0.002
256	0.511 E-13	0.179 E-13	2.47	0.080	23.6	0.005
512	0.870 E-13	0.373 E-13	4.24	0.173	189	0.012
1024	0.178 E-12	0.811 E-13	7.29	0.407	1512 (est.)	0.026
2048	0.942 E-12	0.369 E-12	12.80	0.820	12096 (est.)	0.059

Table 4: Example 4, Numerical Results for Algorithm 4.

5 Conclusions and Generalizations

In this paper we have described a set of four algorithms for computing FFTs for nonequispaced data to any specified precision. An alternative group of algorithms for the problems considered in this paper is presented in [10]. Similarities and differences between the two approaches are summarized below.

1. Both sets of algorithms use a standard FFT.

2. Both sets of algorithms use interpolation formulae to transform function values from equispaced to nonequispaced points and vice-versa. The algorithms of this paper use an interpolation scheme based on the FMM, while the algorithms of [10] use an interpolation scheme based on the Fourier analysis of Gaussian bells.

3. For the application of the linear transformations being considered the algorithms of [10] are the more efficient of the two.

4. For the inversion of these linear transformations, the direct schemes of this paper are generally more efficient than the iterative schemes of [10] whose complexity is dependent on the distribution of the nodes.

5. The FMM-based approach leads to a set of closely related forward and inverse algorithms which can be generalized to complex data points.

In conclusion, a group of algorithms has been presented for the rapid application and inversion of matrices of the Fourier kernel. These problems can be viewed as generalizations of the discrete Fourier transform, and the algorithms, while making use of certain simple results from analysis, are very versatile, and have a broad range of applications in many branches of mathematics, science and engineering.

The results of this paper are currently being applied to problems in a diversity of areas. Examples include problems in the numerical solution of parabolic partial differential equations, the analysis of seismic data, the modelling of semiconductors, weather prediction and the numerical simulation of fluid behavior.

Several obvious generalizations of the results of this paper are discussed below.

1. Problems 1 and 2 involve the evaluation of an N-term series at N points. Straightforward modifications to Algorithms 1 and 2 will allow the efficient evaluation of these N-term series at M points, where $M \neq N$. These modifications have been implemented.

2. The algorithms of this paper are based on a special case of a more general idea, namely the adaptive use of interpolation techniques to speed up large scale computations. Other examples of this approach include the use of wavelets for the construction of fast numerical algorithms (see, for example, [1], [4]), and the use of multipole or Chebyshev expansions for the compression of certain classes of linear operators (see, for example, [2], [6], [16]).

3. One of the more far-reaching extensions of the results of this paper is a set of algorithms for discrete Fourier transforms in two and three dimensions. Detailed investigations into higher dimensional problems of this type are currently in progress and will be reported at a later date.

References

[1] B. Alpert, G. Beylkin, R. Coifman and V. Rokhlin, *Wavelets for the Fast Solution of Second-Kind Integral Equations*, SIAM J. Sci. Stat. Comp., 14 (1993).

[2] B. Alpert and V. Rokhlin, *A Fast Algorithm for the Evaluation of Legendre Expansions*, Technical Report 671, Yale Computer Science Department, 1988.

[3] D. H. Bailey and P. N. Swarztrauber, *The fractional Fourier transform and applications*, SIAM Review, 33 (1991), pp. 389–404.

[4] G. Beylkin, R. Coifman and V. Rokhlin, *Fast Wavelet Transforms and Numerical Algorithms I*, Comm. on Pure and Applied Mathematics, 44 (1991), pp. 141-183.

[5] E. Oran Brigham, *The Fast Fourier Transform and its Applications*, Prentice Hall Inc., Englewood Cliffs, N.J., 1988.

[6] J. Carrier, L. Greengard and V. Rokhlin, *A Fast Adaptive Multipole Algorithm for Particle Simulations*, SIAM J. Sci. Stat. Comp., 9 (1988), pp. 669–686.

[7] J. W. COOLEY AND J. W. TUKEY, *An algorithm for the machine computation of complex Fourier series*, Math. Comp., 19 (1965), pp. 297–301.

[8] G. DAHLQUIST AND A. BJORCK, *Numerical Methods*, Prentice Hall Inc., Englewood Cliffs, N.J., 1974.

[9] A. DUTT, M. GU AND V. ROKHLIN, *Fast Algorithms for Polynomial Interpolation, Integration and Differentiation*, Technical Report 977, Yale Computer Science Dept, 1993.

[10] A. DUTT AND V. ROKHLIN, *Fast Fourier Transforms for Nonequispaced Data*, SIAM J. Sci. Stat. Comp., to appear, 1993.

[11] D. GOTTLIEB, M. Y. HUSSAINI AND S. ORSZAG, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt, D. Gottlieb and M. Y. Hussaini, SIAM, Philadelphia PA, 1984, p.1.

[12] D. GOTTLIEB AND S. ORSZAG, *Numerical Analysis of Spectral Methods*, SIAM, Philadelphia PA, 1977.

[13] I. S. GRADSHTEYN AND I. M. RYZHIK, *Table of Integrals, Series and Products*, Academic Press Inc., 1980.

[14] L. GREENGARD, *Spectral Integration and Two-Point Boundary Value Problems*, Technical Report 646, Yale Computer Science Department, 1988.

[15] L. GREENGARD AND V. ROKHLIN, *A Fast Algorithm for Particle Simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.

[16] V. ROKHLIN, *A Fast Algorithm for the Discrete Laplace Transformation*, Journal of Complexity, 4 (1988), pp. 12-32.

[17] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer Verlag, New York, 1980.

[18] C. VAN LOAN, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.

[19] H. JOSEPH WEAVER, *Theory of Discrete and Continuous Fourier Analysis*, Wiley, 1989.